Article

Quality Prediction and Process Optimization for Silicon Wafer Diamond Wire Sawing Based on Simcenter AI Studio

Lu Zhang[1,*]

[1] School of Mechanical Engineering & Automation, University of Science and Technology Liaoning, Anshan, China
* Correspondence: Lu Zhang, School of Mechanical Engineering & Automation, University of Science and Technology Liaoning, Anshan, China

Abstract: In diamond wire sawing (DWS) of photovoltaic silicon wafers, total thickness variation (TTV) and micro-cracks are difficult to predict online, and process optimization still depends largely on trial and error. This study takes the silicon wafer DWS dataset (solar_raw for experiments, solar_sim for simulation) officially released by the "Lightweight Design and AI Application" track of the national final of the 19th "Gaojiao Cup" National Undergraduate Advanced Graphics Technology and Product Information Modeling Innovation Contest, and carries out cleaning, merging, training, evaluation, and reverse optimization on the Simcenter AI Studio 2025.0.1 platform. Cleaning leaves 141 experimental samples and 152 simulation samples, from which an inner join yields 62 merged samples. On the 62-row merged table, Linear Regression (OLS) in AI Studio reaches a test-set RMSE of 2.775 μm (relative error 9.08%) under a 70/30 split, and the Model Simulator predicts TTV=19.956 μm for a given set of process inputs. A separate Python re-computation (scikit-learn/xgboost, 5-fold cross-validation, 7 regression models) on the 141-row experimental table again finds OLS the most accurate (RMSE=1.136 μm, R^2=0.907), with feed_rate the dominant factor for TTV (r=+0.706); for micro-crack classification, Logistic Regression (LogReg) performs best (F1=0.527, AUC=0.816). A multi-fidelity ablation shows that adding finite-element simulation features (wire stress/deflection) does not improve prediction accuracy for this dataset under the small-sample condition. Reverse optimization on the platform (Find Optimal Input Settings, with three modes Specific Value / Minimize / Maximize) and on the Python side (Ridge surrogate + RF-crack-probability ε-constraint) both recommend the same low-TTV, low-crack process window; the platform's Top1 falls within the Python top-3, and the Top1 window cuts the predicted TTV by about 43.5% relative to the experimental-table mean. The results suggest that a no-code AutoML platform is a practical option for silicon wafer sawing process optimization.

Keywords: diamond wire sawing of silicon wafer; total thickness variation (TTV); micro-cracks; simcenter AI studio; multi-fidelity modeling; process optimization

Received: 05 June 2026
Revised: 16 July 2026
Accepted: 29 July 2026
Published: 31 July 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The thickness uniformity and surface/subsurface damage of photovoltaic silicon wafers directly affect cell conversion efficiency and yield. With its thin wire, low cutting force, and high throughput, diamond wire sawing (DWS) has replaced loose-abrasive sawing as the mainstream slicing process. Even so, total thickness variation (TTV) and micro-cracks are still the main quality bottlenecks: wafers with excessive TTV cause trouble in downstream thickness matching and module lamination, and micro-cracks act as potential fracture sources. The process couples wire tension, feed rate, slurry concentration, and coolant temperature, so trial-and-error tuning rarely finds a globally good process window under multiple objectives and leans heavily on operator experience.

Existing studies mostly focus on experimental investigation and physical modeling: Dong et al examine surface and subsurface damage of monocrystalline silicon cut by electroplated diamond wire [1]; Qiu et al analyze how equipment and tool factors affect the machining performance of endless diamond wire sawing [2]; Li et al study the motion characteristics and periodic saw-mark formation in electroplated diamond wire sawing of monocrystalline silicon wafers [3]; Liu et al investigate the influence of process parameters on the surface quality in micro-shaped-abrasive diamond wire saw slicing of monocrystalline silicon [4]; Wang et al examine the cutting force and surface quality in immersion-fluid-free abrasive-assisted wire sawing of monocrystalline silicon based on the hydrodynamic effect [5]; Singh et al optimize texturisation parameters for PERC-type solar cells produced using diamond wire saw wafers [6]; and Cheng et al review research progress on subsurface micro-crack damage of silicon wafers cut by diamond wire saw. Most of this work draws on a single experimental data source or on experimental/physical modeling [7]; systematic evidence on fusing experimental and simulation data of different fidelities is scarce, and the coding required places such methods beyond the reach of many front-line process engineers. Simcenter AI Studio 2025.0.1 (hereinafter AI Studio; the successor of RapidMiner, which Altair acquired in 2022 and rebranded as Altair AI Studio, and which joined the Siemens Simcenter portfolio after Siemens acquired Altair in 2025) offers a graphical operator-flow environment built around the Process view: data import, cleaning, feature engineering, training, evaluation, and reverse optimization are all draggable operators, and every step stays traceable. Using the contest-released open dataset solar_raw/solar_sim (provenance in Sec. 2), this paper reproduces the full workflow in AI Studio and re-computes it independently with Python 5-fold cross-validation, focusing on three questions: how cleaning and multi-fidelity fusion should be implemented; what limits the achievable TTV/micro-crack prediction accuracy; and whether an interpretable process window can be recommended from such a small sample.

2. Data Understanding and Cleaning

The dataset used in this study is the official open dataset of the "Lightweight Design and AI Application" track (Mechanical category) of the national final of the 19th "Gaojiao Cup" National Undergraduate Advanced Graphics Technology and Product Information Modeling Innovation Contest (Chengtu Contest, organized by the China Graphics Society; the track designates Altair/Simcenter AI Studio as its official AI software), distributed to participating teams by the organizing committee. It provides two CSV tables (snake_case names denote dataset fields): solar_raw (160 experimental rows) records 5 process parameters (wire_speed, wire_tension, feed_rate, slurry_conc, coolant_temp) and 3 quality outputs (surface roughness surface_rough, total thickness variation-field name ttvs, hereinafter uniformly denoted TTV in the text-and the binary micro-crack flag crack_flag∈{0,1}, which serves as the classification target); solar_sim (160 simulation rows) records the same three-key process parameters and 2 finite-element simulation features (wire stress sim_wire_stress, wire deflection sim_deflection; hereinafter "simulation features"). The two tables can be inner-joined by the three keys wire_speed/wire_tension/feed_rate.

Following the given cleaning specification of the dataset, the Filter Examples operator is used in the AI Studio Process view: the experimental table is filtered by rules A-1 (wire_speed≥5), A-2 (ttvs≥0), A-3 (surface_rough not missing, directly removed if absent), A-4 (crack_flag∈{0,1}), reducing it from 160 rows to 141 rows (5 deleted for wire_speed<5, 4 for ttvs<0, 7 for surface_rough missing, 3 for abnormal crack_flag); the simulation table is filtered by an independent Filter operator with rules B-1 (sim_wire_stress≥0) and B-2 (sim_deflection≤0.5), each deleting 4 rows, reducing it from 160 to 152 rows. The Filter rule configuration interface is shown in Figure 1.

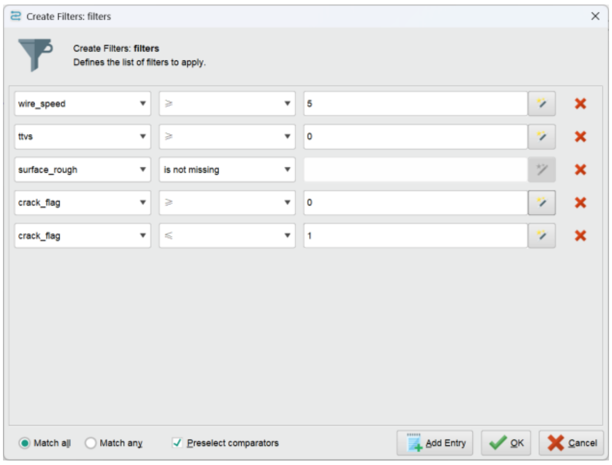

Figure 1. Filter Examples Operator Rule Configuration Interface (5 Filter Entries Covering Cleaning Rules A-1--A-4; Rule a-4 Occupies Two Entries)

3. Multi-Source Fusion and Data Preparation

The cleaned experimental and simulation tables are inner-joined in the Process view via the Join operator by the three keys wire_speed/wire_tension/feed_rate, yielding 62 merged samples that simultaneously contain experimental outputs and simulation features. The left/right key-attribute configuration of the Join operator is shown in Figure 2. Subsequently, the Set Role operator marks the target column ttvs as label and the remaining columns as inputs; Select Attributes chooses the modeling columns; and the Split Data operator splits into training and test sets at 0.7/0.3. The complete data preparation segment (dual-source Retrieve → dual Filter → Join → Set Role → Select Attributes → Split Data) Process-view overview is shown in Figure 3.

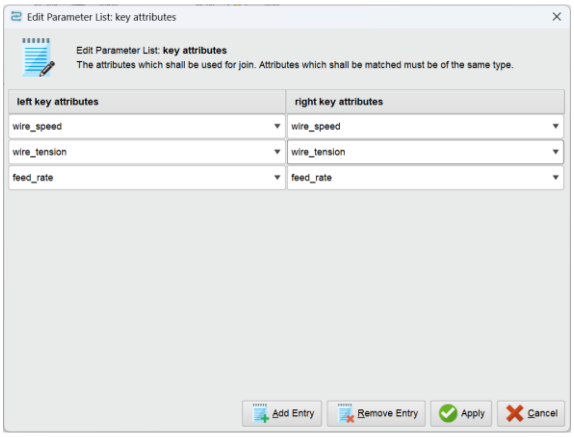

Figure 2. Join Operator Three-Key Key-Attribute Configuration (Left/right Key Attributes)

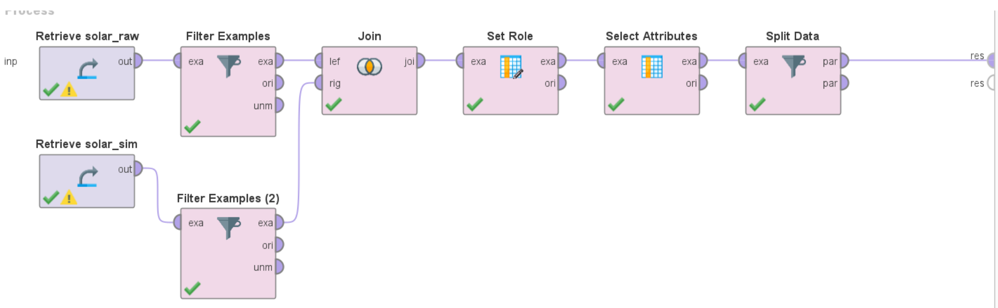

Figure 3. AI Studio Data Preparation Process-view Overview (Retrieve ×2 → Filter Examples ×2 → Join → Set Role → Select Attributes → Split Data)

4. TTV Prediction Modeling

4.1. AI Studio Measurement: Linear Regression and GBT

After Split Data, the Multiply operator copies the training and test sets once each, feeding them to the training operator and the Model Simulator inference operator respectively, and both paths converge at the Performance operator, which outputs RMSE; Figure 4 shows the complete training+evaluation workflow. Under the measured 70/30 split of the 62-row merged table, the Linear Regression (OLS) test-set RMSE=2.775 μm with relative error 9.08%. Swapping the training operator for Gradient Boosted Trees (GBT) while keeping the rest of the structure unchanged, the GBT Performance output under the same split is close to, but slightly behind, that of Linear Regression-the parameter--TTV relationship in this task is largely linear with mild collinearity, leaving little room for complex nonlinear models.

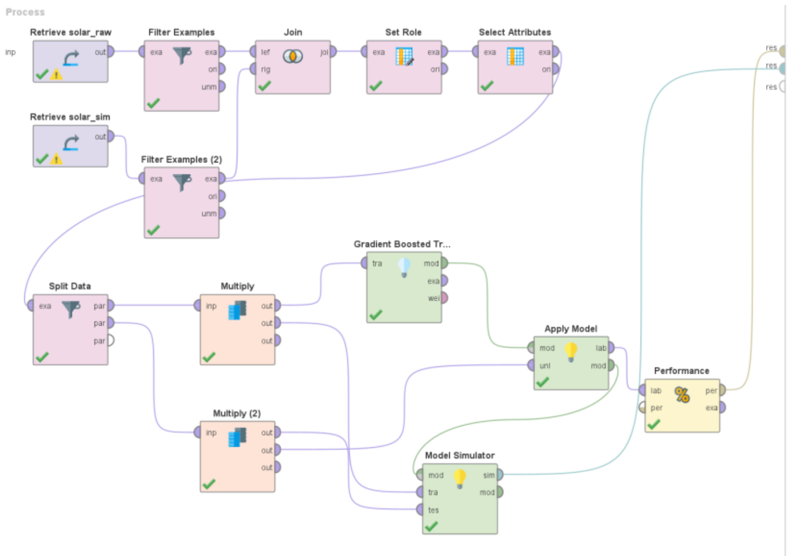

Figure 4. Complete Process View of GBT Training + Evaluation in AI Studio

4.2. What-If Process Optimization (Model Simulator)

The Model Simulator operator runs the trained GBT model on arbitrary process inputs and reports the contribution direction of each feature. Figure 5 shows one such prediction: TTV=19.956 μm; the Important Factors for Prediction panel marks which features support or contradict this prediction, giving a qualitative handle for process tuning.

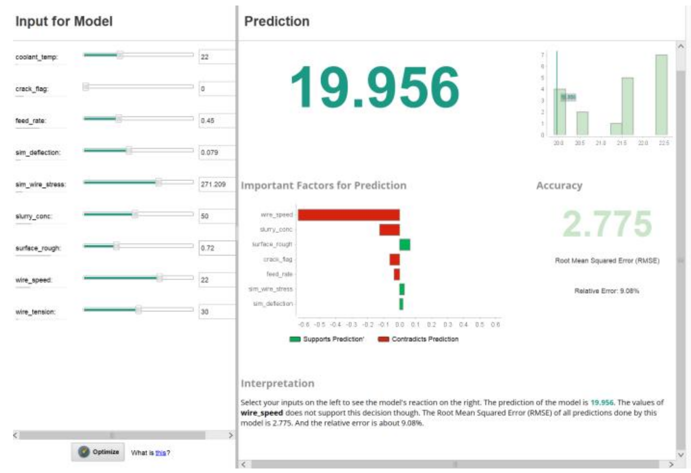

Figure 5. TTV Prediction and Feature Contribution Directions under What-If Input for the GBT Model

4.3. Reverse Optimization on the Platform (Find Optimal Input Settings)

The platform also exposes a dedicated reverse-optimization operator, Find Optimal Input Settings (Figure 6), which sits on top of the same trained GBT model used in 4.2 and inverts the prediction. Given a TTV target and the allowed input range, it searches the input space for the parameter combination that drives the model output to that target. The wizard proceeds in four steps: Define Targets (three directions: Specific Value / Minimize / Maximize), Define Constraints (lower/upper bounds), Optimization Parameters (search algorithm and iteration limits), and Running Optimization (progress and recommendation). The three directions correspond to the three typical process questions; in the dialog shown, Specific Value is selected with target TTV = 20 μm. For cross-validating the Python-side recommendation in Sec. 5, the platform is set to Minimize TTV under the same 5%--95% quantile ranges used in the grid scan, and the platform's Top1 falls within the top-3 of the Python Ridge-surrogate ranking reported in Sec. 5, to within the parameter's discrete grid step. The platform relies on its built-in search, whereas the Python script runs an exhaustive 120,000-candidate scan with an RF-classifier ε-constraint; that both land on the same process window indicates the data-driven recommendation is robust to the choice of optimizer.

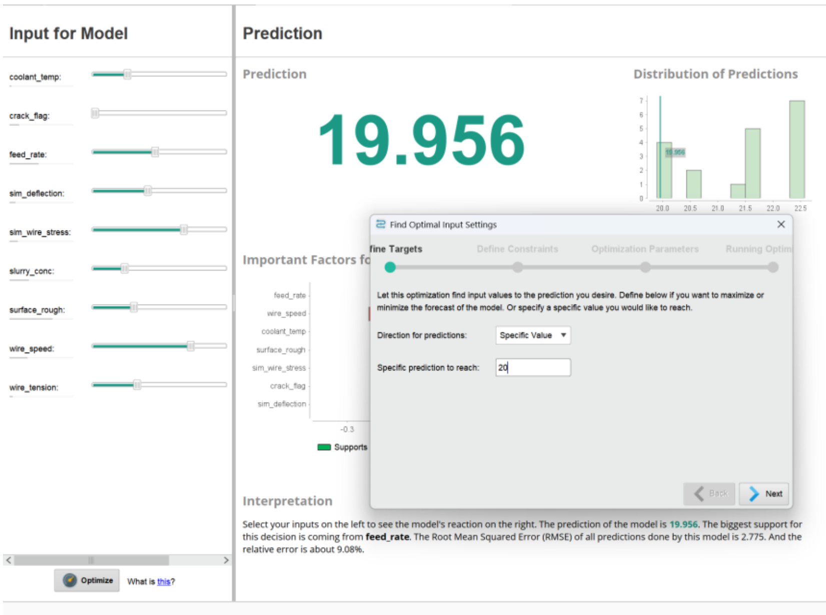

Figure 6. Find Optimal Input Settings Dialog (Direction = Specific Value, Target TTV = 20 Mm Shown; the Dropdown Also Offers Minimize and Maximize Modes)

4.4. Independent Python Re-Computation: 5-Fold CV Horizontal Comparison

The AI Studio conclusions are cross-checked with an independent Python implementation on the cleaned experimental table (n=141). Since the AI Studio measurement in Sec. 4.1 uses the 62-row merged table after the Join, only the qualitative model ranking-not the absolute RMSE-is compared between the two environments. All models are wrapped in a z-score standardization + model pipeline and evaluated with 5-fold cross-validation (random seed 42) over 7 regression models (Table 1); no separate normalization operator is inserted on the AI Studio side, because linear and tree models are invariant to linear feature scaling, and the Python-side standardization mainly keeps scale-sensitive models such as SVR and GPR comparable. OLS and Ridge Regression come out best and second best (CV R^2 of 0.907 and 0.907), matching the AI Studio observation that Linear Regression slightly outperforms GBT; Gaussian Process Regression (GPR) underfits badly ($R^2 \approx 0.265$); its default kernel is apparently ill-suited to the noise level of this task. Random-Forest feature importance and Pearson correlation both point to feed_rate as the dominant factor for TTV (r=+0.706). This agrees with Dong et al [1], whose single-factor/orthogonal experiments found feed rate to have the greatest influence on surface waviness, and is qualitatively in line with the feed-speed/wire-speed ratio--surface quality trend reported by Liu et al; the data-driven and experimental conclusions back each other up [4].

Table 1. Python 5-Fold CV Comparison of 7 Regression Models.

Model	5-fold CV RMSE / μm	5-fold CV R2
Linear Regression (OLS)	1.136	0.907
Ridge Regression	1.137	0.907
Support Vector Regression (SVR)	1.647	0.805
Random Forest (RF)	1.618	0.806
Gradient Boosted Trees (GBT)	1.504	0.834
Gaussian Process Regression (GPR)	3.145	0.265
XGBoost	1.752	0.780

Note: Cleaned Experimental Table N=141, 5 Process Features; Z-Score + Model Pipeline; 5-Fold CV, Seed 42.

5. Multi-Fidelity Ablation and Reverse Process Optimization

Whether adding simulation features to experimental data actually improves TTV prediction is tested with three feature configurations, denoted A, B, and C: A is the 141-row experimental table with the 5 process parameters; B is the 62-row inner-joined merged table restricted to the same 5 parameters (the 2 simulation features are deliberately left out, so that A vs B isolates the pure sample-size effect of the inner join); C is the same 62-row merged table with sim_wire_stress/sim_deflection added (B vs C isolates the pure feature effect). The same 7 models are trained with 5-fold CV under each configuration; Figure 7 compares the RMSE. Two points stand out. First, from A to B the RMSE of the four nonlinear models (SVR/RF/GBT/XGBoost) rises while that of OLS/Ridge/GPR drops slightly-the sample loss of the inner join (141→62) mainly hurts model classes that need more data. Second, C minus B is ≥0 for all 7 models (average +0.106 μm): the marginal gain from simulation features is unstable or negative, so the common default that "introducing finite-element simulation features necessarily improves prediction accuracy" does not hold here-a finding for the silicon wafer sawing dataset adopted in this study, not a general claim about small-sample scenarios.

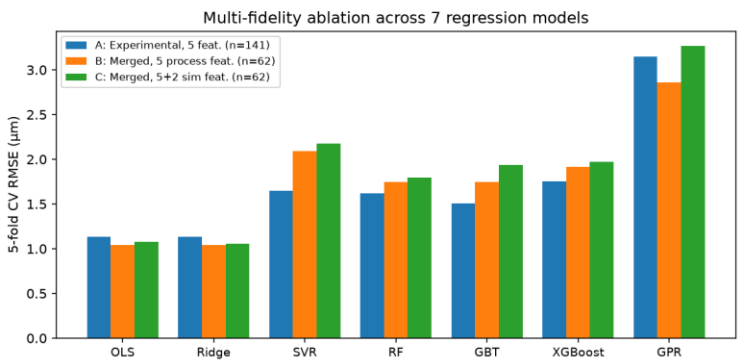

Figure 7. Multi-fidelity Ablation: 5-Fold CV RMSE Comparison of All 7 Regression Models under Configurations a / B / C (a: Experimental N=141, 5 Process Features; B: Merged N=62, Same 5 Features; C: Merged N=62, 5+2 Simulation Features).

For the micro-crack classification target crack_flag, 3 classification models are compared with 5-fold CV (Table 2); Logistic Regression (LogReg) is clearly the best of the three (F1=0.527, AUC=0.816), and RF feature importance again ranks feed_rate and wire_speed as the two dominant variables. The reverse optimization uses an ε-constraint single-objective design instead of weighted multi-objective aggregation. Within the 5%--

95% quantile range of each process parameter in the experimental table, 120,000 candidate points are sampled uniformly (seed 42). Candidates whose micro-crack probability, predicted by the class-weight-balanced RF classifier, exceeds 0.3 are screened out; the threshold sits above the positive-class base rate (19.9%) so that the feasible set is not emptied while high-risk candidates are still removed. The remaining candidates are then ranked by the TTV predicted with a Ridge surrogate model-Ridge is near-optimal in CV accuracy and, unlike tree models, extrapolates smoothly over a dense candidate set (the GBT of Sec. 4.2 only serves as the platform-side single-point What-If demonstration). Table 3 lists the results: the Top1 window lowers the predicted TTV by about 43.5% against the mean of the 141-row experimental table (21.322 μm), and its crack probability of ≈0.005 is the resulting value of that candidate, not the constraint threshold. The window offers a quantitative starting point for process tuning; for micro-crack damage mechanisms see also the review by Cheng et al [7].

Table 2. 5-Fold CV Comparison of 3 Micro-Crack Classification Models.

Model	Accuracy	Precision	Recall	F1	AUC
Logistic Regression (LogReg)	0.731	0.415	0.760	0.527	0.816
Random Forest (RF)	0.773	0.455	0.507	0.464	0.729
XGBoost	0.816	0.627	0.427	0.480	0.746

Note: 120,000 Candidates Uniformly Sampled in the 5%--95% Quantile Ranges; RF Crack Probability ≤ 0.3 as E-Constraint; Ranked by Ridge-predicted TTV.

Table 3. Top-3 Recommended Process Windows from Reverse Optimization Based on the Ridge Surrogate Model.

wire_speed	wire_tension	feed_rate	slurry_conc	coolant_temp	Pred TTV/μm	Crack prob
27.10	20.60	0.149	65.82	20.25	12.04	0.005
26.57	17.05	0.175	64.33	16.97	12.04	0.072
27.02	21.27	0.144	62.15	18.68	12.17	0.007

Note: Experimental Table N=141; Positive Class Crack_flag=1, 28/141≈19.9%; Class-Weight Balanced; Seed 42.

6. Conclusions and Innovations

Working on the open dataset solar_raw/solar_sim released by the 19th Chengtu Contest ("Lightweight Design and AI Application" track, national final), this study carries the full cleaning--merging--training--evaluation--reverse-optimization workflow through the Process view of Simcenter AI Studio 2025.0.1 and re-computes it independently with Python 5-fold CV. The main findings: (1) cleaning leaves 141 experimental and 152 simulation samples, and the inner join yields 62 merged samples; micro-cracks account for 19.9% of the experimental table (28 of 141), an imbalanced binary classification problem. (2) In AI Studio (70/30 split), Linear Regression reaches a test-set RMSE of 2.775 μm (relative error 9.08%), and the Model Simulator predicts TTV=19.956 μm. (3) In Python 5-fold CV, OLS and Ridge are the most accurate (R^2=0.907/0.907) and Logistic Regression leads the classification task (F1=0.527), consistent with the AI Studio results. (4) The multi-fidelity ablation gives C−B ≥0 for all 7 models: with samples this scarce, experimental data should dominate, and simulation data should be fused via inner join only with caution. (5) The reverse optimization recommends a process window with low TTV and low crack probability; the Top1 window cuts the predicted TTV by about 43.5% relative to the experimental-table mean, with a crack probability of ≈0.005. Taken together, a no-code AutoML platform proves workable for silicon wafer sawing process optimization, and the workflow provides a reproducible reference for similar small-sample multi-fidelity tasks.

This work differs from prior studies in four respects:

6.1. Platform-Based, Auditable, and Reproducible Modeling Workflow

Where most prior studies rely on bench experiments and physical modeling, the entire workflow here lives in the Process-view graphical operator flow of Simcenter AI Studio 2025.0.1 (Retrieve→Filter→Join→Set Role→Select Attributes→Split→train/evaluate→Model Simulator). Cleaning, fusion, modeling, evaluation, and reverse optimization each remain on the platform as a traceable operator chain, which sidesteps the reproducibility and audit problems of script-based experiments.

6.2. "Experimental + Simulation" Dual-Source Data Fusion and Multi-Fidelity Ablation within a Unified Platform

The experimental table solar_raw (141 rows after cleaning) and the finite-element simulation table solar_sim (152 rows after cleaning) are fused via a three-key inner join into 62 samples, and a multi-fidelity ablation over 7 models yields a simulation-feature increment $\Delta RMSE \geq 0$ for every model-adding finite-element simulation features buys no prediction gain for this dataset and process scenario under the small-sample condition. The result runs against the usual expectation and, within the scope of this study, warns against assuming that multi-fidelity fusion automatically improves accuracy.

6.3. Closed Loop from Prediction to Process Optimization: Dual-Route Reverse Optimization on the Platform and in Python

On the platform side, the Model Simulator operator runs the trained GBT model on arbitrary process inputs and reports each feature's contribution direction (measured What-If prediction: TTV = 19.956 μm). On top of it, the Find Optimal Input Settings operator inverts the prediction through the platform's built-in search, offering three modes (Specific Value / Minimize / Maximize) that answer the three typical process questions-what settings produce a target TTV, what settings minimize TTV, what settings maximize it. On the Python side, a 120,000-candidate exhaustive grid scan is then run over the same 5%--95% quantile ranges, with a class-weight-balanced RF classifier enforcing the ε-constraint that the predicted micro-crack probability must not exceed 0.3; the remaining candidates are ranked by a Ridge surrogate model. The two routes converge to the same window: the platform's Top1 falls within the Python top-3, and the recommended Top1 reduces TTV from the experimental-table mean of 21.322 μm to 12.04 μm, a drop of about 43.5%, with a predicted crack probability of ≈0.005. Prediction and optimization thus close into one loop for the continuous indicator TTV, with crack probability as the constraint.

6.4. Quality Evaluation Covering Both Continuous and Discrete Indicators within One Workflow

One and the same workflow covers both types of quality indicator: a regression model predicts the continuous indicator TTV (in the Python re-computation, OLS leads the 7 models in 5-fold CV, RMSE=1.136 μm, R^2=0.907), and a Logistic Regression classifier predicts the discrete micro-crack indicator (best F1=0.527). The two models are trained and evaluated separately but share the same cleaning--fusion--CV pipeline, so the evaluation spans continuous and discrete quality indicators at the workflow level.

Data availability: The solar_raw/solar_sim dataset is the official open dataset of the "Lightweight Design and AI Application" track of the 19th "Gaojiao Cup" Chengtu Contest national final, obtainable from the contest organizing committee (chengtudasai.com); all cleaning rules, random seeds, split ratios, and pipeline settings used in this study are stated in the text to ensure reproducibility.

References

1. H. Dong, Y. Gao, and C. Yang, "Surface and subsurface microstructure properties of monocrystalline silicon cut by electroplated diamond wire saw," *Surf. Sci. Technol.*, vol. 3, p. 30, 2025, doi: 10.1007/s44251-025-00098-1.
2. J. Qiu, Z. Wang, C. Liu et al., "Investigation on the equipment and tool influencing factors on machining performance of endless diamond wire sawing," *Int. J. Adv. Manuf. Technol.*, vol. 121, pp. 6341–6359, 2022, doi: 10.1007/s00170-022-09705-z.
3. F. Li, S. Zhu, Y. Gu et al., "Motion characteristic and periodic saw mark formation in electroplated diamond wire sawing of monocrystalline silicon wafers," *J. Mech. Sci. Technol.*, vol. 40, pp. 5265–5273, 2026, doi: 10.1007/s12206-026-0636-8.

4. Z. Liu, M. Ge, Q. Wang et al., "Experimental investigation and analysis of the influence of process parameters on the surface quality of micro-shaped abrasive diamond wire saw slicing," *Int. J. Adv. Manuf. Technol.*, vol. 144, pp. 1895–1909, 2026, doi: 10.1007/s00170-026-17981-2.
5. Y. Wang, J. Li, Y. Li et al., "Study on the cutting force and surface quality in immersion fluid free abrasive assisted wire sawing monocrystalline silicon based on hydrodynamic effect," *Int. J. Adv. Manuf. Technol.*, vol. 143, pp. 3665–3684, 2026, doi: 10.1007/s00170-026-17628-2.
6. D. Singh, B. K. Pant, V. Bhardwaj et al., "Optimization of Texturisation Parameters for Enhanced Efficiency of PERC Type Solar Cell Produced Using Diamond Wire Saw Wafers," *J. Inst. Eng. India Ser. D*, 2025, doi: 10.1007/s40033-025-00859-0.
7. D. Cheng, Y. Gao, and C. Yang, "Research progress on subsurface microcrack damage of silicon wafer cut by diamond wire saw: a review," *J. Mater. Sci.: Mater. Electron.*, vol. 36, p. 1977, 2025, doi: 10.1007/s10854-025-16034-w.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Publisher and/or the editor(s). Publisher and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.